VILNIUS UNIVERSITY

FACULTY OF MATHEMATICS AND INFORMATICS

MATHEMATICS AND MATHEMATICAL APPLICATIONS COURSE

SATELLITE IMAGE RECOGNITION TASK

1st Report

Author: Edvardas Timoscenka 3rd year student of Mathematics and Mathematical Applications

Vilnius

2023

Contents

1. What objects do you plan to recognize in satellite imagery?

My goal for this project is to perform semi-supervised semantic segmentation on high-resolution satellite images from the dataset described in part 2. For semantic segmentation, I chose to keep most of the classes from the dataset, presented in Table 1.

Table 1: Classes used for the task.

Class	% pixels
Urban fabric	9.6 %
Industrial, commercial, public, military, private and transport units	6.4 %
Mine, dump, and construction sites	0.7 %
Artificial non-agricultural vegetated areas	1.1 %
Arable land (annual crops)	29.5 %
Permanent crops	1.0 %
Pastures	29.0 %
Forests	15.9 %
Herbaceous vegetation associations	4.6 %
Open spaces with little or no vegetation	0.4 %
Wetlands	0.7 %
Water	1.0 %

2. What images do you plan to analyze?

The dataset I chose for this task is called MiniFrance [1]. The dataset consists of high-resolution satellite images of different regions of France. To make the original data computationally tractable, I plan to create a smaller dataset from the MiniFrance dataset called tinyMiniFrance, which consists of around 3,500 (1.7%) of the original data. tinyMiniFrance would be created by uniform sampling over each region, thus keeping the original class distribution of the dataset.

3. What computing resources do you plan to use?

For this task, I have decided to use a personal computer with Nvidia GeForce 4090 GPU 24GB VRAM, and 11th Gen Intel Core i9 CPU with 32GB RAM.

4. **What programming language (frameworks, libraries) do you plan to use?**

   To perform the task, I am going to use: Python and its' PyTorch, PyVision, Caffe libraries, and CUDA.

5. **Aspects you think are important (Neural Network architecture choice)**

   To perform a comparative analysis, I plan to use PyTorches' implementation of DeepLabV3 [2] architecture with ResNet-50 [3], ResNet-101 [3], and MobileNet-v3 [4] backbones, in addition to the PyTorches' model, I plan to compare results to U-NET [5] architecture and try to implement Xception [6] as a backbone to DeepLabV3 to see, if any improvements could be made this way.

6. **If the program code has already been presented, whether you plan to improve it or to create a new one.**

   For the Pytorches' implementation of the DeepLabV3 model, I do not plan to change any code in libararies implementation. For the U-NET model, I plan to write code with the same architecture as in the original paper and for Xcode do the same and apply that code to the DeepLabV3. For teaching the model and data processing, I plan to use my code.

7. **Bibliography**

[1] J. Castillo-Navarro, B. Le Saux, A. Boulch, N. Audebert, and S. Lefèvre, "Semi-supervised semantic segmentation in Earth Observation: the MiniFrance suite, dataset analysis and multi-task network study," *Mach. Learn.*, vol. 111, no. 9, pp. 3125–3160, Sep. 2022, doi: 10.1007/s10994-020-05943-y.

[2] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking Atrous Convolution for Semantic Image Segmentation." arXiv, Dec. 05, 2017. Accessed: Sep. 17, 2023. [Online]. Available: http://arxiv.org/abs/1706.05587

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition." arXiv, Dec. 10, 2015. Accessed: Sep. 17, 2023. [Online]. Available: http://arxiv.org/abs/1512.03385

[4] A. Howard *et al.*, "Searching for MobileNetV3." arXiv, Nov. 20, 2019. Accessed: Sep. 17, 2023. [Online]. Available: http://arxiv.org/abs/1905.02244

[5] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation." arXiv, May 18, 2015. Accessed: Sep. 17, 2023. [Online]. Available: http://arxiv.org/abs/1505.04597

[6] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI: IEEE, Jul. 2017, pp. 1800–1807. doi: 10.1109/CVPR.2017.195.